Principles of programming - course description

General information

Course name	Principles of programming
Course ID	11.3-WE-AutP-PrinProgr-Er
Faculty	Faculty of Computer Science, Electrical Engineering and Automatics
Field of study	Automatic Control and Robotics
Education profile	academic
Level of studies	First-cycle Erasmus programme
Beginning semester	winter term 2021/2022

Course information

Semester	1
ECTS credits to win	5
Course type	obligatory
Teaching language	english
Author of syllabus	• dr inż. Grzegorz Łabiak

Classes forms

The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Lecture	30	2	-	-	Exam
Laboratory	30	2	-	-	Credit with grade

Aim of the course

The goal of the subject is to teach programmining in C language. In the first place the learning consists in teaching syntax and semantics of C language. Next, students are instructed how to implement simple computational algorithms. Apart from that some elements of computational complexity are introduced, which allows to asses computational cost of implemented solutions.

Prerequisites

Fundamentals of mathematics

Scope

Introductory information: a program and its components
Pogramming environment. Source files. Compilation. Basic elements of program and its structure. Main function. Functions and procedures.
Basic types. Variables. In/out operations. Operators.
Iterative loops: for, while, do-while
Decision instructions.
Creation of own Functions.
Arrays and character strings.
Structures.
Pointers.
Sorting algorithms: buble sort, selection sort, quick sort.

Teaching methods

Lecture, laboratory exercises

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
Student knows the difference between structural and object oriented programming		• an ongoing monitoring during classes	• Laboratory
Student is able to define real world problemes and present them as computational problmes, critialy analise solutions and assess computational complexity		• a quiz • an ongoing monitoring during classes	• Laboratory
Student is able to implement typical algorithms, such as sorting, searching, etc.		• an evaluation test	• Lecture • Laboratory

Outcome description	Outcomesymbols	Methods of verification	The class form
Student is able to solve algorithmic problem, implement it in C language, run and test it.		• a quiz • an ongoing monitoring during classes	• Laboratory

Assignment conditions

Lecture - exam - in order to get a credit it is necessary to pass all of the required tests (oral or written)
Laboratory - the main condition to get a pass are sufficient marks for all exercises and tests conducted during the semester
Calculation of the final Grade: lecture 50% + laboratory 50%

Recommended reading

1. Kernighan B. W., Ritchie D. M.: Język Ansi C, WNT, Warszawa, 1994.
2. Sielicki A.: Laboratorium programowania w języku Pascal, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 1994

Further reading

1. Aho A. V., Hopcroft J. E., Ullman J. D.: Projektowanie i analiza algorytmów, Helion, Warszawa, 2003.
2. Banachowski L., Diks K., Rytter W.: Algorytmy i struktury danych, WNT Warszawa, 2001.
3. Roszkowski J.: Analiza i projektowanie strukturalne, Helion, Gliwice, 2002.
4. Wirth N.: Algorytmy + struktury danych = programy, WNT, Warszawa,1989.

Notes

Modified by dr hab. inż. Wojciech Paszke, prof. UZ (last modification: 12-07-2021 07:56)